Bootable El-Torito CD with GRUB Legacy

From OSDev Wiki

This tutorial guides you through making a bootable CD .iso image with GRUB Legacy. We are going to create an El-Torito "no-emulation" bootable CD, which is different from a bootable CD emulating a floppy disc.

Difficulty level

Beginner

Contents

Requirements

You will need the following:

- mkisofs, which is in fact superseeded by genisoimage.
- A (Multiboot compliant) kernel that GRUB can boot.
- The El-Torito GRUB Legacy stage2 file, called `stage2_eltorito`.

Ubuntu/Debian

In Ubuntu or Debian you can install the required software like this:

```
sudo apt-get install genisoimage grub
```

We need a place to store the files on the CD image:

```
cd                                  # Go to your home directory.
mkdir -p isofiles/boot/grub
```

You now created an `isofiles` directory in your home folder and a `boot/grub` sub-folder inside that. We need the stage2 file. It is installed from the grub package. Just copy it:

```
cp /usr/lib/grub/i386-pc/stage2_eltorito  ~/isofiles/boot/grub/
```

Windows

Look at this article about how to install and use mkisofs. We need a place to store the files on the CD image. Create a folder `isofiles`, which contains the sub-folder `boot\grub` needed by GRUB. We need the stage2 file. You can get it by downloading a package (http://www.geocities.com/imulgrew/grub_files.tar.gz) and unpack it using your favourite unpacking application. Find the file `stage2_eltorito` and copy it to `isofiles\boot\grub`.

Install a kernel

Get a kernel of your own choosing and copy it to wherever you like inside the `isofiles` folder. Preferably it should be placed in the `boot` sub-folder. Now create a `menu.lst` file. This file controls what menu entries GRUB should provide and how the kernels are booted. It must be placed in the GRUB folder `boot/grub` and contain something like this:

```
default 0
#timeout 30

#title Boot from hard disk
#chainloader (hd0)+1

title My kernel
kernel /boot/kernel-file    # Edit it to the filename of your ke
```

In GRUB there's a concept called root. It's the disk drive or partition where you access the files, like the kernel image and modules. When booting from CD using `stage2_eltorito` you don't need to set the root as it is already set to "`(cd)`".

Create the .iso image

In the following I use the command `genisoimage`, but you can change it to `mkisofs` if that is what its called on your system. Open a command prompt/terminal and go to where the `isofiles` folder is located. It is your home directory on Ubuntu. Issue the command:

```
genisoimage -R -b boot/grub/stage2_eltorito -no-emul-boot -boot-
            -boot-info-table -o bootable.iso isofiles
```

Now you have a file called `bootable.iso`. Test it using your favourite emulator or burn it to a CD and test on a real computer. I will just explain the command line arguments we used for genisoimage:

-R	Use the Rock Ridge protocol, which enables lower-case filenames on the CD. This is needed by GRUB.
-b file	The file to boot (the filename is in the created ISO 9660 file system).
-no-emul-boot	Enables no emulation El-Torito boot.
-boot-load-size 4	Specifies the number of 512-bytes sectors to load. Four 512-byte sectors (2048 bytes) is one CD sector and is the number supported by most BIOS.
-boot-info-table	Patches the boot file to contain info about the CD image. It's needed by GRUB.
-o bootable.iso	The filename of the resulting .iso image.
isofiles	Other arguments are the files and folders that should be included on the CD. In this case it's only the content of isofiles.

Advanced stuff

Give it a label

You want to label your CD image, so you can later recognize it when loading your CD. Just pass a **-V** command line argument to genisoimage followed by the name you want.

Make it be quiet

Some day you probably want to create an .iso image directly in your Makefile. If you don't like the output that genisoimage creates, just pass the command line argument **-quiet**.

Sometimes you get a warning about the input character set used. If that is a problem just pass the argument **-input-charset ascii** (or utf8 if that is what you use on your filesystem).

Make the image from different files and folders

When you are building your kernel the parts on the .iso image could be located at different paths. Imagine your project files are like this:

```
project/
├── build/
│   ├── modules/
│   │   ├── iso9660.mod
```

```
|   |   └── kbdrv.mod
|   └── my-kernel.elf
├── src/
|   ├── grub/
|   |   └── menu.lst
|   └── ...
├── thirdparty/
|   └── grub/
|       └── stage2_eltorito
└── Makefile
```

You don't want to copy those files around in order to create the .iso image. Just use the **-graft-points** argument, like this:

```
genisoimage -graft-points # ...other arguments here...
            boot/my-kernel.elf=build/my-kernel.elf
            boot/grub/menu.lst=src/grub/menu.lst
            boot/grub/stage2_eltorito=thirdparty/grub/stage2_elt
            modules/=build/modules/
```

You can use both files and folders together with the -graft-points argument.

Solving problems

If you don't find a solution to your problem below, please ask in the forums (http://forum.osdev.org/) .

Mkisofs says Uh oh, I cant find the boot image

The output of mkisofs looks like this:

```
mkisofs: Uh oh, I cant find the boot image 'isofiles/boot/grub/stage2_eltorito' !
```

This problem arises because mkisofs/genisoimage looks for its boot image as a subdirectory of the filesystem on the CD; make sure that the path you specify starts with 'boot/' rather than the name of your ISO directory ('isofiles/' in this example).

Source: this thread

I get a permission denied error in Linux

The output of genisoimage looks like this:

```
Size of boot image is 4 sectors -> No emulation
genisoimage: Permission denied. Error opening boot image file 'isofiles/boot/grub/stage2_eltorit
```

The problem is that the -boot-info-table command line argument patches the boot file to include information about the CD, but the boot file is not writable by genisoimage. GRUB needs the boot file to be patched in order to access the CD properly. The solution is to make it writable, like this:

```
chown myself isofiles/boot/grub/stage2_eltorito     # Make sure y
                                                    # with your u
chmod u+w isofiles/boot/grub/stage2_eltorito        # Make the fi
```

See Also

Articles

- mkisofs
- Bootable CD - using floppy disk emulation
- GRUB 2 - for info on using GRUB 2 to make a bootable CD

Forum

- Thread about making your own .iso image from scratch without tools

External Links

- Package containing the GRUB stages (http://www.geocities.com/imulgrew/grub_files.tar.gz) (including `stage2_eltorito`)
- Source for GRUB Legacy (ftp://alpha.gnu.org/gnu/grub/grub-0.97.tar.gz) to build it yourself

Categories: Level 1 Tutorials | Tutorials | Bootloaders
